

REMARKS/ARGUMENTS

Claims 1-22 and 24-25 have been canceled without disclaimer or prejudice. New claims 26-41 have been added. These claims are supported, for example, by pages 12-29 of the specification. No new matter has been added.

Objection to the Drawings

In the Office Action, the Examiner objected to Fig. 2. A proposed formal drawing of Fig. 2 has been submitted for Examiner's approval. Fig. 2 illustrates a simple example of a JSP. As noted in the specification, the example shows the response page, which is intended to be a short list of the day of the month and year, at the moment the request is received by the server. The page itself contains some fixed template text, and JSP elements that are shown underlined in the figure. The underlined actions are executed on the server side. When a client makes a request, such as an HTTP request, a request object requests a response from the JAVASERVER™ container. The first element creates a JAVA™ Bean named clock, of type calendar.jspCalendar. The next two elements use the Bean to display some of its properties (i.e. month and year). The output is sent to a response object which sends a response back to the client.

A JSP page is executed by a JSP container, which is installed on a Web server, or on a Web enabled application server. The JSP container delivers requests from a client to a JSP page and responses from the JSP page to the client. JSP pages may be implemented using a JSP translation or compilation phase that is performed only once, followed by a request processing phase that is performed once per request. The translation phase creates a JSP page implementation class that implements a servlet interface.

Typically, a JSP page contains declarations, fixed template data, action instances that may be nested, and scripting elements. When a request is delivered to a JSP page, all these components are used to create a response object that is then returned to the client. As with standard Web Pages, JSP pages may contain "tags." A tag is a textual element within a document that provides instructions for formatting or other actions. For example, World Wide Web documents are set up using HTML (Hyper-Text Mark-up Language) tags which serve various functions such as controlling

the styling of text and placement of graphic elements, and also providing links to interactive programs and scripts. (Specification, paragraphs 4-7.)

It is respectfully requested that Fig. 2 illustrates the interaction between the client and a JSP page that is running in a container. Accordingly, it is respectfully requested that the Examiner withdraw this objection.

Objection to the Specification

As suggested by the Examiner, the trademark Java has been accompanied by generic terminology. As is known in the art, JAVA™ is a programming language expressly designed for use in the distributed environment of the Internet. It was designed to have the "look and feel" of the C++ language, but it is simpler to use than C++ and enforces an object-oriented programming model. JAVA™ can be used to create complete applications that may run on a single computer or be distributed among servers and clients in a network. It can also be used to build a small application module or applet for use as part of a Web Page. Applets make it possible for a Web Page user to interact with the page. Accordingly, it is respectfully requested that the Examiner withdraw this objection to the specification.

Rejection of claims

In the Office Action, the Examiner has rejected claim 23 under 35 U.S.C. 102(b) as being anticipated by Andrew C. Staugaard, Jr., "Structured and Object-Oriented Techniques, An Introduction Using C++," 1997, Prentice Hall, Inc. (hereinafter *Staugaard*). (Office Action, page 7, paragraph 14). It is respectfully submitted that the Examiner has not addressed several of the features that are recited in claim 23. The recited features of claim 23 that have not been addressed include: (a) generating and storing code for any attribute values in a tag instance, (b) instantiating the tag, (d) creating a BodyEvaluation object if necessary, and (g) synchronizing any indicated scripting variables. As such, it is respectfully submitted that the rejection of claim 23 is improper and should be withdrawn.

Furthermore, it is respectfully submitted that the cited art, among other things, does not teach or suggest providing a tag-handler object for a custom action tag. The tag handler provides a run-time representation of the custom action tag and includes a do-start method for processing of a start-tag, a do-body method for processing of the body, and a do-end method for processing of the end-tag of the custom action tag.

Still further, it is earnestly believed that the cited art does not teach or suggest invoking the do-start method, invoking the do-body method and processing, by the do-body method the body of the custom action tag to translate the body from a first scripting language to platform independent code that can be executed to perform actions intended by the custom action tag. As such, the cited art cannot teach or suggest determining by the do-body method whether further processing is required to translate the body from a first scripting language to platform independent code that can be executed to perform the actions intended by the custom action. Nor does the cited art teach or suggest invoking the do-end method of the tag-handler object when the do-body method determines that no further processing is required.

Based on the foregoing, it is submitted that all pending claims are patentably distinct over the cited art of record. Additional limitations recited in the independent claims or the dependent claims are not further discussed as the above-discussed limitations are clearly sufficient to distinguish the claimed invention from the cited art. Accordingly, it is respectfully requested that the Examiner withdraw all rejections.

Applicant believes that all pending claims are allowable and respectfully requests a Notice of Allowance for this application from the Examiner. Should the Examiner believe that a telephone conference would expedite the prosecution of this application, the undersigned can be reached at the telephone number set out below.

If there are any issues remaining which the Examiner believes could be resolved through either a Supplemental Response or an Examiner's Amendment, the Examiner is respectfully requested to contact the undersigned attorney at the telephone number listed below.

Applicants hereby petition for an extension of time which may be required to maintain the pendency of this case, and any required fee for such extension or any further fee required in connection with the filing of this Amendment is to be charged to Deposit Account No. 500388 (Order No. SUN1P253).

Respectfully submitted,
BEYER WEAVER & THOMAS, LLP



R. Mahboubian
Reg. No. 44,890

P.O. Box 778
Berkeley, CA 94704-0778
(650) 961-8300

APPENDIX A

APPENDIX B

FIGURE 2
MARKED-UP VERSION

Client

HTTP

Request

Response

Container



JSP Page

```
<html>
<jsp:useBean id="clock"
              class="calendar.jspCalendar" />
<ul>
<li>Day: <%=clock.getDayOfMonth() %>
<li>Year: <%=clock.getYear() %>
</ul>
</html>
```

Figure 2

APPENDIX C

SUBSTITUTE ABSTRACT
CLEAN VERSION FOR ENTRY

SUBSTITUTE ABSTRACT
MARKED-UP VERSION

ABSTRACT

A multi-lingual tag extension mechanism suitable for use with the ~~JavaServer~~
JAVASERVERTM Pages platform. Rather than creating a closure abstraction, the present
invention “in-lines” a body evaluation for the tags. A doStart () method processes a start tag
and determines if a body needs to be evaluated. If so, a body evaluation buffer is passed to a
5 doBody () method for body evaluation. Once the body evaluation is complete, a doEnd ()
method completes the processing by synchronizing the variables. The scripting details of the
present invention match any nesting of the tag, so the at the structure corresponding to the
original scripting is preserved. Furthermore, the present invention is not dependent on the
specifics of the scripting language used to form the Web page.